Model-based Testing

Ed Brinksma
University of Twente, The Netherlands

This course will provide an overview of model-based testing and its applications. The approach that we present was developed over the past two decades and uses the formal framework of concurrency theory based on labelled transition systems and formalizations of observable behaviour as have been developed in the context of process algebra and Petri net theory. The theory has proven to be successful as a basis for the automatic generation and evaluation of test cases from formal specifications, especially for control-intensive applications, such as embedded systems. As part of the course we will present a test derivation and execution tool, and discuss some real-life application projects.

The course consists of four lectures.

In the first lecture, we provide a general introduction to testing in general, and model-based testing in particular. We introduce basic testing terminology and concepts, and relate our approach to other model-based approaches in testing. We also introduce the part of concurrency theory that we will need to develop our testing framework. Good, classical introductions to this theory can be found in Milner [8] and Hoare [7]. A seminal article relating concurrency theory to testing is De Nicola & Hennessy [6]. A useful collection of older work on testing and testing theory have been collected in Brinksma & Tretmans [5].

In the second lecture we present the basic theory for the derivation of functional tests from models in the form of input/output transition systems. An important feature of the theory is that absence of response by an implementation is treated as a special kind of output, known as quiescence. We demonstrate the test derivation and excution tool TorX that is based on this theory. TorX supports so-called on-line test derivation, in which the continuation of a test case are calculated during its execution. A good reference to the basic theory for test derivation can be found in Tretmans [10]. An updated version of the theory is given in Timmer, Brinksma & Stoelinga [9]. An overview of testing tools is provided in Belinfante, Frantzen & Schallhart [1].

In the third lecture is about testing and real-time. We show how both theory and tools can be extended to deal with real-time behaviour in specifications, implementations and tests. We will contrast pragmatic approaches with fully formal ones, and discuss the relative merits. The theoretical extension can be found in Brandán Briones & Brinksma [3], a tool in Bohnenkamp & Belinfante [2].

In the fourth, and final lecture we discuss the problem of test selection and coverage. In almost all realistic cases testing for all possible errors will require a very large or even infinite set of tests that cannot be used for practical purposes. This leads to the problem of selecting tests and the related issue of how to express and measure the quality of selected test cases. The main reference here is Brandán Briones, Brinksma & Stoelinga [4].

References

1. A. Belinfante, L. Frantzen, C. Schallhart. *Tools for Test Case Generation.* LNCS 3472, Springer, pp. 391-438; 2005.
2. H. Bohnenkamp, A. Belinfante. *Timed Testing with TorX.* LNCS 3582, Springer, pp. 173-188; 2005.

3. L. Brandán Briones, E. Brinksma. *A Test Generation Framework for Quiescent Real-Time Systems*. LNCS 3395, Springer, pp. 64-98; 2004.
extended version available at `http://eprints.eemcs.utwente.nl`

4. L. Brandán Briones, E. Brinksma, M. Stoelinga. *A Semantic Framework for Test Coverage*. LNCS 4218, Springer, pp. 399-414; 2006.
extended version available at `http://eprints.eemcs.utwente.nl`

5. E. Brinksma, G. Tretmans. *Testing Transition Systems: An Annotated Bibliography*. LNCS 2067, Springer, pp. 187-195; 2001.

6. R. De Nicola, M. Hennessy. *Testing Equivalence for Processes*. Theoretical Computer Science, Vol. 34, pp. 83-133; 1984.

7. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall; 1985.

8. R. Milner. *Communication and Concurrency*.Prentice Hall; 1989.

9. Timmer, M., Brinksma, E., Stoelinga, M. *Model-Based Testing*. In: Software and Systems Safety, Proc. of the *International Summer School Marktoberdorf 2010*, Vol. 30, IOS-Press 2011, pp. 1-32; 2011

10. G. Tretmans. *Test Generation with Inputs, Outputs and Repetitive Quiescence*. Software-Concepts and Tools, Vol. 3, pp. 103-120; 1996.